Probing Functional Correctness in Diffusion Language Models

Guan-Ming Chiu
National Taiwan University
gmchiu@arbor.ee.ntu.edu.tw

Jeng-Yue Liu
Carnegie Mellon University
buffettl@andrew.cmu.edu

Abstract

Diffusion language models generate text by iteratively denoising all tokens in parallel, but when and where their hidden states encode whether the output will be functionally correct remains unknown. We present the first probing study of DLM internals, training linear classifiers on hidden states to predict functional correctness. Across two models (LLaDA-8B, Dream-7B) and four tasks, we find that DLMs uniquely accumulate correctness signal across denoising steps (AUC gains of 0.08–0.11 on reasoning tasks), absent in single-pass AR decoding. However, step-0 signal reflects prompt difficulty rather than diffusion-specific computation. Signal emergence is task-dependent: structural tasks show flat profiles while reasoning tasks show gradual buildup. The two models exhibit distinct layer dynamics, with LLaDA concentrating signal in upper layers while Dream redistributes toward lower layers. We further show that probe confidence can identify likely failures, enabling selective generation that avoids 36–98% of wasted compute.

1 Introduction

Diffusion language models (DLMs) have emerged as a promising alternative to autoregressive (AR) generation (Austin et al., 2021a; Sahoo et al., 2024; Lou et al., 2024). Unlike AR models, which generate tokens sequentially, DLMs produce all output tokens simultaneously through iterative denoising: starting from a fully masked or noisy sequence and progressively refining it over many steps.

This parallel generation process raises a fundamental question: *when* during denoising, and *where* within the model's layers, does the network encode whether its output will be functionally correct? In AR models, probing classifiers have revealed how linguistic and factual knowledge localizes across layers and positions (Tenney et al., 2019; Geva et al., 2023; Azaria and Mitchell, 2023). However, no analogous study exists for diffusion language models, whose bidirectional, iterative generation process creates a fundamentally different representational landscape.

We address this gap with the first probing study of DLM hidden states. Our contributions are: (1) DLMs uniquely accumulate correctness signal across iterative denoising, a capacity absent in single-pass AR decoding: while step-0 hidden states already carry above-chance signal comparable to an AR probe on the same prompt (and thus largely attributable to prompt difficulty rather than diffusion), DLMs gain a further 0.08–0.11 AUC through subsequent denoising on reasoning tasks; (2) we identify task-dependent emergence patterns, with structural tasks showing flat AUC profiles from step 0 and reasoning/code tasks requiring iterative refinement; and (3) we observe distinct layer dynamics between LLaDA and Dream, with LLaDA concentrating signal in upper layers while Dream redistributes signal toward lower layers on simple tasks. Code and data are available at https://github.com/guan404ming/dllm-probing.

2 Background

Diffusion language models. LLaDA (Nie et al., 2025) uses block masking during training and generates by progressively unmasking tokens over a fixed schedule. Dream (Ye et al., 2025) employs a global masking schedule with a different unmasking strategy. Both models condition on a fully visible prompt and iteratively denoise a generation region of masked tokens.

Probing classifiers. Probing involves training simple classifiers on frozen model representations to test whether specific information is encoded (Belinkov, 2022; Conneau et al., 2018). We adopt this methodology for DLMs, probing hidden states at intermediate denoising steps.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop), pages 163–172

Related work. Prior studies have probed AR model representations for linguistic structure (Tenney et al., 2019), factual knowledge (Geva et al., 2023), and truthfulness (Azaria and Mitchell, 2023). Confidence-based methods for AR models use early-exit thresholds to skip computation (Schuster et al., 2022; Elbayad et al., 2020). Two concurrent works study DLM representations from complementary angles. Shnaidman et al. (2025) demonstrate activation steering for safety behaviors in masked DLMs, showing that DLM representations are amenable to linear intervention. Wang et al. (2025) identify temporal oscillation in DLM outputs, where correct answers appear mid-denoising but are overwritten in later steps. Mündler et al. (2025) propose constrained decoding with context-free grammars to enforce syntactic correctness in DLM outputs; our work instead asks whether the model's own representations already encode whether the output will be correct, without external constraints. We complement these prior findings by probing hidden representations for functional correctness across the full layer-step grid, characterizing *when* and *where* correctness signal emerges.

3 Method

Hidden state extraction. Given a DLM with L layers, we run 128-step denoising and extract hidden states at 7 checkpoint steps $t \in \{0, 1, 4, 16, 32, 64, 127\}$. At each checkpoint, we obtain the full hidden state tensor of shape $(L, \text{gen\_length}, d)$. We partition the generation region into 4 equal-length position regions and mean-pool within each region, yielding features of shape $(L, 4, d)$ per instance per step. We choose 4 regions to align with LLaDA's block length of 32 tokens (gen_length / block_length = 4 for JSON schema's 128-token blocks) while keeping the number of features manageable given the limited sample sizes. An ablation over 1, 2, and 4 regions shows best AUC varies by less than 0.02 across configurations (Appendix G).

Probe training. For each (step, layer, region) triple, we reduce dimensionality with PCA to 64 components, apply standard scaling, and train a logistic regression classifier to predict functional correctness (Pedregosa et al., 2011). We evaluate with 5-fold stratified cross-validation, reporting AUC as our primary metric. Control probes trained on shuffled labels (3 random permutations, max AUC across the same layer-region grid) yield 0.53–0.58 across all eight (dataset, model) settings, well below the actual probe AUCs of 0.65–0.88 (dashed grey trace in Figure 2); the slight offset above 0.50 reflects multiple-comparison selection across layers and folds, and is uniform across configurations.

Functional correctness. For JSON schema validation, an output is correct if it parses as valid JSON and matches the reference structure. For math reasoning (GSM8K), an output is correct if the extracted numeric answer matches the ground truth. For code generation (MBPP), an output is correct if the generated function passes all provided test assertions. For science QA (ARC), an output is correct if the extracted answer letter matches the gold label.

Selective generation. We simulate instance-level filtering by training a per-step probe (using the best layer and region from the full analysis). Let p be the probe's predicted probability of functional correctness and define confidence as $\max(p, 1 - p)$. At inference time, if the probe predicts $p < 0.5$ with confidence exceeding a threshold τ, we skip the remaining denoising steps for that instance entirely, producing no output rather than wasting compute on a generation predicted to be incorrect. We report the fraction of instances filtered and the probe's classification accuracy.

4 Experimental Setup

Models. We evaluate LLaDA-8B-Instruct (33 layers) (Nie et al., 2025) and Dream-7B-Instruct (29 layers) (Ye et al., 2025), two open-weight DLMs with different masking strategies.

Datasets. We select four tasks spanning structural to reasoning-heavy complexity, testing whether iterative denoising contributes differently by task type: (1) **JSON schema validation** (272 instances from `eth-sri/json-mode-eval-extended`, an extension of `NousResearch/json-mode-eval` that augments the original 100-instance benchmark with additional schema variants); each instance pairs a JSON Schema with a natural-language request, and a generation is functional if it parses as JSON and matches the reference structure exactly; (2) **GSM8K** (1,319 test instances) (Cobbe et al., 2021), multi-step math reasoning; (3) **MBPP** (257 sanitized test instances) (Austin et al., 2021b), code generation verified by execution; and (4)

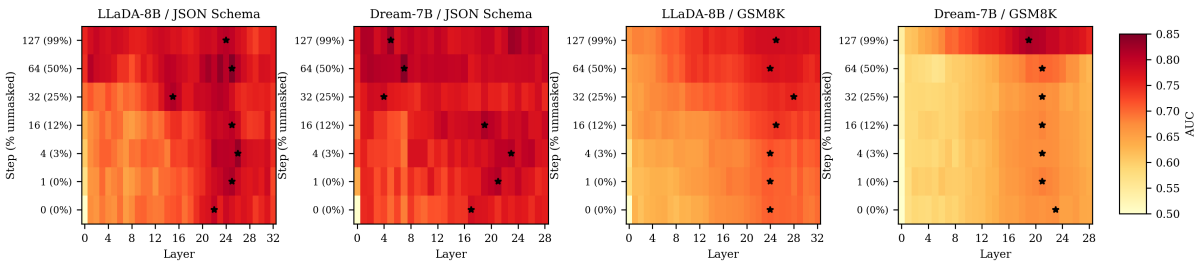

Figure 1: AUC heatmaps (layer × step) for JSON schema and GSM8K. Each cell shows the best AUC across the 4 position regions for that (layer, step) pair. Stars mark the best layer per step. JSON schema shows strong signal from step 0; GSM8K shows gradual emergence. Note Dream's layer migration on JSON schema. Per-cell standard deviations across 5-fold CV are ≤ 0.05. MBPP and ARC heatmaps in Appendix B.

	JSON	GSM8K	MBPP	ARC
LLaDA-8B	48.5%	66.3%	40.9%	78.2%
Dream-7B	46.0%	61.5%	44.4%	74.5%

Table 1: Baseline functional correctness rates (seed=0, 128 denoising steps).

ARC-Challenge (1,172 test instances) (Clark et al., 2018), multiple-choice science QA with short answers that isolate reasoning from generation length. Generation lengths are 256, 512, 256, and 256 tokens, respectively.

Baseline functional rates. Table 1 shows the functional correctness rates with 128 denoising steps. All datasets have reasonable class balance, enabling meaningful probe training.

5 Results

5.1 Task-Dependent Emergence

Figure 1 shows AUC heatmaps for JSON schema and GSM8K (MBPP and ARC in Appendix B). Two distinct emergence patterns are visible.

For **JSON schema**, correctness signal is already strong at step 0 (AUC $\approx$ 0.78–0.80 $\pm$ 0.04) and remains relatively flat across all denoising steps. This indicates that the prompt encoding alone, before any denoising occurs, already contains signal predictive of structural correctness.

For **GSM8K**, step-0 signal is already well above chance (AUC $\approx$ 0.67–0.71 $\pm$ 0.03), but denoising brings substantially more gain: AUC rises to $\approx$ 0.78–0.82 $\pm$ 0.03 by step 127 (Figure 2).

MBPP follows the same gradual-buildup pattern as GSM8K: step-0 AUC is $\approx$ 0.75–0.76 $\pm$ 0.04 and peaks at $\approx$ 0.84–0.87 $\pm$ 0.05 at later steps (step 64 for LLaDA, step 127 for Dream).

ARC-Challenge shows the weakest and most denoising-dependent signal: step-0 AUC is $\approx$ 0.61–0.67, and signal builds gradually to $\approx$ 0.71–0.75 by step 64–127. Dream's ARC signal is particularly flat until a sharp jump at the final step (0.61 $\rightarrow$ 0.71), suggesting science reasoning requires nearly complete denoising before correctness is predictable.

All four tasks have above-chance step-0 signal, but JSON schema gains only ΔAUC $\approx$ 0.04 from denoising while GSM8K, MBPP, and ARC gain $\approx$ 0.08–0.11, reflecting the difference between structural and reasoning tasks. Standard deviations across 5-fold CV are ≤ 0.05 for all configurations, indicating stable estimates. Because DLMs process all generation positions jointly via bidirectional attention, correctness-relevant information is available across the full output region from step 0.

We also tested whether the probe could select better random seeds: scoring 5 seeds per instance at step 1 and running full denoising only for the top-scoring seed yields +0.0% improvement for both models on JSON schema, confirming that the step-0/1 signal reflects instance-level difficulty rather than seed-specific trajectory quality.

5.2 Divergent Layer Dynamics

The heatmaps in Figure 1 reveal distinct layer dynamics between LLaDA and Dream (full per-step best layers in Table 3, Appendix C).

LLaDA concentrates its best probing signal consistently in upper layers (L22–28) across JSON schema, GSM8K, and ARC, similar to patterns observed in AR transformers. MBPP is an exception, where best layers are more variable (L15–31).

Dream shows a task-dependent migration pattern. On JSON schema, the best layer migrates from the upper range (L17–23 at steps 0–16) to lower layers (L4–7 at steps 32–127). On GSM8K,

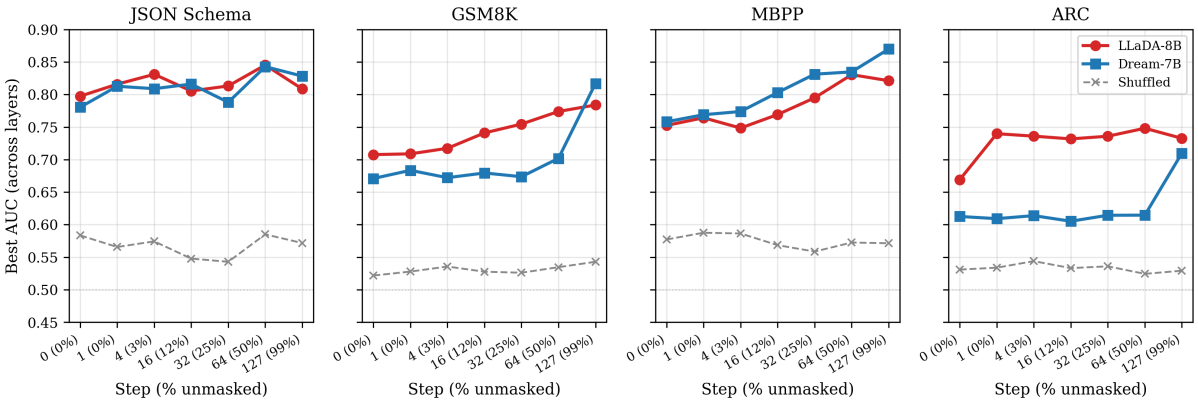

Figure 2: Best AUC across layers at each denoising step (x-axis annotated with cumulative % of generation tokens unmasked). JSON schema is flat (≈0.80 from step 0), while GSM8K, MBPP, and ARC show gradual emergence. Dashed grey trace is the shuffled-label control (mean AUC across both models, averaged over 3 random label permutations); the dotted line marks chance (0.50). Standard deviations across 5-fold CV are ≤ 0.05 for all probe points.

MBPP, and ARC, however, Dream's best layers remain in the mid-to-upper range (L15–25) throughout denoising, with no downward migration. We discuss a possible interpretation in terms of Dream's AR pretraining origin in §6.

5.3 Filtering Hopeless Instances (Offline)

We evaluate whether per-step probe confidence can identify instances whose outputs will be incorrect, enabling a hypothetical system to skip generation entirely for hopeless instances. This is an *offline* simulation on hidden states already extracted from full 128-step generation, intended to characterize the upper bound of what the probe can offer; it does not account for the cost of running the probe in a live inference loop. Table 2 shows results with a confidence threshold of $\tau = 0.80$.

The results are strongly task-dependent. For JSON schema and MBPP, both models save over 95% of compute and clear the majority baseline by 7–22 percentage points: the probe is confident from step 0, identifying most instances predicted to fail immediately. For GSM8K and ARC, savings are more modest (36–83%) because reasoning tasks require more denoising steps before the probe can confidently distinguish hopeless instances. For ARC, where the baseline functional rate exceeds 74% (Table 1), classification accuracy is only marginally above the majority baseline, indicating limited discriminative value on this task.

Comparison to confidence heuristics. A natural concern is whether a hidden-state probe is necessary at all: the model already exposes a per-token softmax distribution, whose entropy or max probability could in principle serve the same role. We re-run the same filtering simulation using two such non-probe baselines (mean entropy and mean max-prob over masked positions in the generation region) and a logistic regressor trained on those scalar features (LR-entropy). Across all eight (dataset, model) settings, the probe outperforms the best raw signal in classification accuracy at $\tau = 0.80$ by 5–20 percentage points, and its peak per-step AUC exceeds LR-entropy by 0.05–0.18 on average (full tables in Appendix J). A second control trains a logistic regression directly on output-text features (length, format markers, and ARC predicted letter); this shallow probe is also outperformed by

	JSON	GSM8K	MBPP	ARC
LLaDA-8B				
Compute saved	**98.4%**	61.3%	95.5%	**82.9%**
Clf. accuracy	73.5%	**73.5%**	66.5%	79.6%
Majority acc.	51.5%	66.3%	59.1%	78.2%
Dream-7B				
Compute saved	96.7%	36.0%	**97.8%**	61.4%
Clf. accuracy	72.4%	**73.5%**	70.0%	75.9%
Majority acc.	54.0%	61.5%	55.6%	74.5%

Table 2: Filtering simulation ($\tau = 0.80$). *Compute saved*: fraction of denoising steps avoided by skipping instances the probe predicts will fail (no output is produced for skipped instances). *Clf. accuracy*: fraction of instances for which the probe correctly predicts functional vs. non-functional. *Majority acc.*: trivial baseline that always predicts the majority class. Lift over majority is large for JSON (+19–22%) and MBPP (+7–15%) but only marginal for ARC (+1.4% on Dream).

the hidden-state probe in 7 of 8 settings with an average AUC gap of 0.12 and a maximum of 0.25 on GSM8K-Dream (Appendix K).

6 Discussion

Why does step-0 signal exist? At step 0, the DLM has performed a forward pass over the prompt with masked outputs, producing representations that correlate with eventual correctness. For structural tasks like JSON generation, this correlation is strong: the representation already contains signal predictive of correctness. For reasoning and code tasks, step-0 signal is weaker, and representations are iteratively refined through denoising. The DLM step-0 signal spans all generation positions simultaneously, arising from bidirectional attention over prompt and masked tokens. An AR baseline probe on Qwen-2.5-7B's last prompt token achieves comparable AUC on most tasks (Appendix F), suggesting that much of the step-0 signal reflects prompt difficulty detectable by any model. On JSON schema, DLM step-0 AUC modestly exceeds the AR probe (0.78–0.80 vs. 0.74); on other tasks, the gap is negligible (Appendix F, Table 6). The distinctive advantage of DLMs lies not in step-0 signal strength but in the capacity to gain additional signal through denoising, a capacity absent in single-pass AR decoding entirely.

Relationship to temporal oscillation. Wang et al. (2025) observe that DLM outputs can oscillate during denoising: correct answers appear at intermediate steps but are overwritten later. Our probe measures whether hidden states *encode* correctness, not whether the current surface output is correct. On JSON schema, hidden-state signal is flat from step 0 while surface tokens are still being refined, suggesting that hidden-state signal predictive of correctness is established early and oscillation reflects surface-level instability rather than changes in the underlying representation.

Layer dynamics and training origin. LLaDA (trained from scratch) shows stable upper-layer localization (best layers L22–28 of 33) resembling AR models, while Dream (fine-tuned from Qwen-2.5) exhibits layer migration on simple tasks but not complex ones. On GSM8K, MBPP, and ARC, Dream's best layers stay in L15–25 (of 29), tracking Qwen-2.5's own probe best layers, which fall in L18–27 (of 28) on the same tasks (Appendix F). On JSON schema, however, Dream's best layers migrate down to L4–7 by step 32, deviating sharply from Qwen-2.5's range. This pattern is consistent with the hypothesis that diffusion fine-tuning preserves the AR base's upper-layer hierarchy on tasks that demand reasoning capacity, but unlocks alternative routing to lower layers when the task is structurally simple enough that the inherited hierarchy is no longer load-bearing. We caution that LLaDA's block-based and Dream's global linear schedules unmask different positions even at matched cumulative fractions, so step-axis comparisons (Figures 1, 2) reflect approximate denoising progress rather than identical model states.

7 Conclusion

We present the first probing study of DLM internals. Our central finding is that DLMs uniquely accumulate correctness signal through iterative denoising (a 0.08–0.11 AUC gain on reasoning tasks), a capacity absent in single-pass AR decoding. The step-0 hidden state already carries above-chance signal (AUC 0.61–0.80), but this is comparable to a probe on an AR model conditioned on the same prompt, indicating that step-0 signal reflects prompt difficulty rather than diffusion-specific computation. We additionally observe task-dependent emergence and distinct layer dynamics between LLaDA and Dream, and show that an offline filtering simulation based on per-step probe confidence avoids 36–98% of wasted compute and outperforms confidence heuristics derived from the model's own softmax distribution. Length-stratified controls confirm the signal is not merely a length confound: AUC drops by 0.00–0.09 after length matching, with MBPP showing no drop at all.

Future work includes validating filtering in a real inference pipeline (with the probe forward pass costed in), scaling to larger DLMs, and exploring probe-guided adaptive compute allocation. Nonlinear or multi-layer probes may capture richer structure.

Limitations

Our probe is a simple logistic regression classifier with PCA-reduced features; nonlinear probes or multi-layer probes might reveal richer structure. The selective generation analysis is a simulation on states from full 128-step generation and does not account for inference overhead of running probes in real time. The AUC gain we observe through denoising (about 0.08–0.11 on reasoning tasks)

cannot be cleanly attributed to diffusion-specific computation versus generic iterative refinement: a single-pass AR model has no analogous comparison point, and we do not test multi-pass AR variants (e.g., self-consistency over multiple samples) that might also accumulate signal across passes.

We study only two models (LLaDA-8B, Dream-7B), both discrete masked diffusion LMs, and four tasks. Continuous diffusion LMs (e.g., Diffusion-LM, GENIE) operate on continuous embedding spaces with different denoising dynamics and are out of scope for this study; whether our findings extend to that family is an open question. Broader coverage across discrete masked architectures (e.g., MDLM, SEDD) and model scales is also needed. LLaDA (block-based) and Dream (global linear) reach approximately the same cumulative fraction of unmasked tokens at each checkpoint step (Figures 1, 2 are labeled accordingly), but the spatial distribution of unmasked tokens differs (LLaDA fills block-by-block, Dream picks most-confident globally), so cross-model comparisons should be read as approximate denoising-progress alignments rather than identical model states. Our interpretation of LLaDA vs. Dream layer dynamics in terms of training origin (§6) is correlational rather than causal: the two models also differ in pretraining data, tokenizer, and architecture, and a controlled fine-tuning ablation (e.g., diffusion fine-tuning the same AR base with and without lower-layer routing capacity) would be needed to causally establish the role of AR pretraining. A length control probe achieves higher AUC than the correctness probe in most settings (Appendix E); after length matching, correctness AUC drops modestly but remains well above chance. Prompt-length-matched probes similarly show only 0.02 average AUC drop (Appendix I). A shallow-feature probe trained jointly on output length, format markers, and ARC predicted letter is outperformed by the hidden-state probe in 7 of 8 settings (Appendix K); the exception is ARC-Dream where the shallow probe narrowly leads, consistent with the marginal probe lift over majority on ARC.

Probe selection (best layer and region) uses the same cross-validation splits as evaluation. A nested CV reanalysis (Appendix H) shows AUC drops by only 0.01 on average (max 0.07), confirming that the main patterns are not artifacts of selection bias. The filtering simulation in §5.3 treats the probe as free at inference time; deploying the same procedure live would incur additional cost from a forward pass at each candidate exit step plus probe evaluation, and would require calibrating the threshold τ on a held-out development split rather than via cross-validation as we do here.

Acknowledgements

We thank Modal for the compute credits that supported the experiments in this work.

References

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Danny Tarlow, and Rianne van den Berg. 2021a. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*, volume 34, pages 17981–17993.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. 2021b. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

Amos Azaria and Tom Mitchell. 2023. The internal state of an LLM knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 967–976.

Yonatan Belinkov. 2022. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136.

Maha Elbayad, Jiatao Gu, Edouard Grave, and Michael Auli. 2020. Depth-adaptive transformer. In *International Conference on Learning Representations*.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. 2023. Dissecting recall of factual associations in auto-regressive language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12216–12235.

Aaron Lou, Chenlin Meng, and Stefano Ermon. 2024. Discrete diffusion modeling by estimating the ratios of the data distribution. *arXiv preprint arXiv:2310.16834*.

Niels Mündler, Jasper Dekoninck, and Martin Vechev. 2025. Constrained decoding of diffusion LLMs with context-free grammars. *arXiv preprint arXiv:2502.11727*.

Shen Nie, Fengqi Zhu, Chao You, Xiaojun Zhang, and Zhenguo Ou. 2025. LLaDA: Large language diffusion with mAsking. *arXiv preprint arXiv:2502.09992*.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. 2024. Simple and effective masked diffusion language models. *arXiv preprint arXiv:2406.07524*.

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Q Tran, Yi Tay, and Donald Metzler. 2022. Confident adaptive language modeling. In *Advances in Neural Information Processing Systems*, volume 35, pages 17456–17472.

Andrey Shnaidman, Philip Marek, Eyal Karpas, and Jonathan Herzig. 2025. Activation steering in discrete diffusion language models. *arXiv preprint arXiv:2505.20389*.

Ian Tenney, Dipanjan Das, and Eleni Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601.

Ziming Wang, Jiahao Luo, Haowei Tang, and Tian Gao. 2025. Time is a feature: Temporal oscillation in discrete diffusion language models. *arXiv preprint arXiv:2505.11229*.

Jiacheng Ye, Jiahui Gong, Yanru Lin, Ting Hin Zhuo, Tong Chen, Xin Jiang, Zhenguo Li, Qun Liu, and Lingpeng Liu. 2025. Dream: Diffusion reasoning with enhanced accuracy and mastery. *arXiv preprint arXiv:2504.16915*.

A Generation and Probing Details

Denoising procedure. Both models run 128 denoising steps. LLaDA uses block-based unmasking: the generation region is divided into blocks of 32 tokens, and each block is denoised sequentially over $128/(\text{gen\_length}/32)$ steps per block. Within each block, the model selects the most confident tokens to unmask at each step. Dream uses a global linear schedule: at step i, the fraction of tokens to unmask is determined by a linearly decreasing timestep schedule from $t = 1$ to $t = \epsilon$, and the most confident masked tokens are unmasked globally.

Prompt construction. For JSON schema, we prepend a system prompt containing the target schema and append a JSON code-fence marker as a generation prefix. For GSM8K, the system prompt instructs the model to solve step-by-step and end with #### followed by the numeric answer. Both models use their respective chat templates. A complete prompt example for each task is shown in Figure 3.

JSON schema (LLaDA chat template)

```
[BOS] system\n You are a helpful assistant that answers in JSON. Here's the JSON schema you must adhere to:\n <schema>\n {schema} \n </schema>\n user\n {user input}\n assistant\n ```json\n
```

GSM8K (Dream chat template)

```
[BOS] im_start system\n Solve the math problem step by step. End your answer with #### followed by the final numeric answer. im_end\n im_start user\n {question} im_end\n im_start assistant\n
```

Figure 3: Schematic prompt examples after each model's chat template is applied (special tokens shown as plain words for readability; the actual tokens used are model-specific control tokens). The generation region (128–512 masked tokens) is appended after the final assistant marker. Both models share the same system/user content; only the wrapping tokens differ.

Hidden state extraction. At each of the 7 checkpoint steps $\{0, 1, 4, 16, 32, 64, 127\}$, we run a forward pass with `output_hidden_states=True` and extract the full hidden state tensor across all layers. The generation region is partitioned into 4 equal-length position regions, and we mean-pool within each region, producing a feature vector of dimension d (4096 for LLaDA, 3584 for Dream) per layer per region per instance.

Probe pipeline. For each (step, layer, region) configuration, we construct the feature matrix $X \in \mathbb{R}^{n \times d}$ and apply: (1) `StandardScaler` (per-feature zero-mean and unit-variance normalization, with mean and variance fitted on the training fold and applied to the held-out fold), (2) PCA reduction to 64

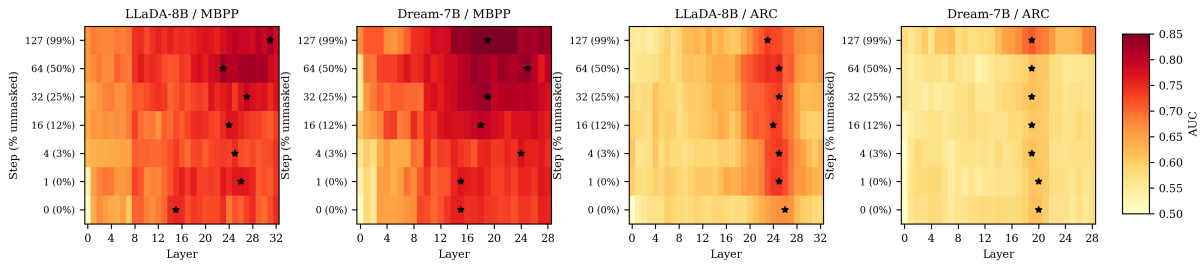

Figure 4: AUC heatmaps for MBPP and ARC-Challenge. Both show gradual emergence similar to GSM8K. ARC has the weakest overall signal (AUC ≤ 0.75).

components (also fitted on the training fold), (3) logistic regression with L2 regularization ($C = 1.0$, LBFGS solver, max 1000 iterations). We use 5-fold stratified cross-validation (random state 42) and report the mean AUC across folds.

Computational resources. All generation and feature extraction use NVIDIA A100 (80GB) GPUs on Modal, with 8 parallel A100s per experiment. Models are loaded in bfloat16 precision. Probe training runs on CPU. Total compute: approximately 30 A100-hours across all experiments.

B Additional Heatmaps

Figure 4 extends the heatmaps to MBPP and ARC. Per-model best-layer ranges (LLaDA MBPP L15–31; ARC L18–28 for both models) confirm that the JSON-schema downward migration in §5.2 does not occur on these tasks. ARC's heatmaps are visibly washed out (most cells ≤ 0.65), consistent with the high majority-class baseline ($\geq 74\%$) limiting headroom.

C Best Layer per Step

Table 3: Best layer (by AUC) at each denoising step. Bold values highlight Dream's migration to lower layers on JSON schema. Other tasks show no such migration.

Step	0	1	4	16	32	64	127
JSON Schema							
LLaDA	22	25	26	25	15	25	24
Dream	17	21	23	19	**4**	**7**	**5**
GSM8K							
LLaDA	24	24	24	25	28	24	25
Dream	23	21	21	21	21	21	19
MBPP							
LLaDA	15	26	25	24	27	23	31
Dream	15	15	24	18	19	25	19
ARC-Challenge							
LLaDA	26	25	25	24	25	25	23
Dream	20	20	19	19	19	19	19

D Selective Generation Details

The selective generation simulation works as follows. For each instance, we iterate through the checkpoint steps in order. At each step, we compute the probe's confidence as $\max(p, 1-p)$ where p is the predicted probability of functional correctness. If the probe confidently predicts failure (confidence exceeds threshold τ and $p < 0.5$), we skip the remaining denoising steps for that instance, producing no output. If no checkpoint triggers skipping, the instance proceeds to full generation. Compute savings are computed as $1 - \frac{t_{\text{skip}}+1}{128}$, where t_{skip} is the step at which filtering occurs.

The main paper reports results at $\tau = 0.80$. Table 4 shows results across a range of thresholds for all configurations.

Table 4: Selective generation results across thresholds (%). "Sv" is compute saved by skipping instances predicted to fail. "Ac" is classification accuracy.

τ	**JSON**		**GSM8K**		**MBPP**		**ARC**	
	Sv	Ac	Sv	Ac	Sv	Ac	Sv	Ac
LLaDA-8B								
0.60	99.2	64.0	83.4	67.2	99.1	63.4	98.6	78.1
0.70	99.0	69.5	74.2	70.3	98.2	65.8	94.1	79.4
0.80	98.4	73.5	61.3	73.5	95.5	66.5	82.9	79.6
0.90	93.3	75.0	34.1	74.2	90.8	68.1	57.0	79.9
Dream-7B								
0.60	98.8	63.2	85.3	56.9	99.1	72.0	97.2	74.8
0.70	97.9	69.1	66.3	62.5	98.8	70.0	86.1	75.7
0.80	96.7	72.4	36.0	73.5	97.8	70.0	61.4	75.9
0.90	87.3	74.6	6.2	74.8	94.9	73.5	24.2	76.8

E Length Control Probe

We train a control probe (same pipeline) to predict binary output length (above/below median) instead of correctness. To control for the confound, we re-train the correctness probe on a length-matched subsample where functional and non-functional instances have identical length distributions (binned matching, 10 quantile bins).

Table 5: Best AUC for correctness, length control, and length-matched correctness probes. After controlling for length, correctness AUC drops modestly but remains well above chance.

	Corr.	Length	Corr. (matched)
JSON Schema			
LLaDA	0.85	0.98	0.81
Dream	0.84	0.99	0.75
GSM8K			
LLaDA	0.79	0.86	0.75
Dream	0.82	0.87	0.80
MBPP			
LLaDA	0.84	0.78	0.84
Dream	0.87	0.80	0.87
ARC			
LLaDA	0.75	0.96	0.75
Dream	0.71	0.98	0.72

F AR Baseline Probe

We probe Qwen-2.5-7B-Instruct (Dream's AR base model) using the last prompt token hidden state. Sub-A predicts DLM labels; Sub-B uses Qwen's own greedy outputs.

Table 6: AR vs. DLM step-0 probe AUC.

	JSON	GSM8K	MBPP	ARC
Qwen Sub-A (DLM)	0.74	0.69	0.76	0.67
Qwen Sub-B (AR)	0.77	0.77	0.69	0.74
LLaDA step-0	0.78	0.71	0.76	0.67
Dream step-0	0.80	0.71	0.76	0.61

G Spatial Pooling Ablation

We pool the 4 extracted regions into 1 (global average), 2 (pairwise average), or 4 (concatenation) groups and re-run the probe.

Table 7: Best AUC by number of position regions. Spread is less than 0.03 within each configuration.

	1 region	2 regions	4 regions
JSON Schema			
LLaDA	0.848	0.854	0.840
Dream	0.846	0.828	0.847
GSM8K			
LLaDA	0.784	0.788	0.784
Dream	0.818	0.814	0.812
MBPP			
LLaDA	0.842	0.860	0.834
Dream	0.864	0.868	0.856
ARC-Challenge			
LLaDA	0.747	0.740	0.744
Dream	0.710	0.711	0.716

H Nested Cross-Validation

To verify that probe selection does not inflate reported AUC, we rerun all experiments with nested cross-validation: an inner 3-fold CV on training data selects the best (layer, region), and the outer 5-fold evaluates on held-out data.

Table 8: AUC drop from nested vs. standard CV. Average drop is 0.01; all qualitative patterns are preserved.

	Avg Drop	Max Drop
JSON Schema		
LLaDA	0.005	0.050
Dream	0.035	0.072
GSM8K		
LLaDA	0.009	0.019
Dream	0.008	0.014
MBPP		
LLaDA	0.025	0.053
Dream	0.021	0.038
ARC-Challenge		
LLaDA	0.002	0.015
Dream	0.005	0.031

I Prompt Length Control

We re-train correctness probes on subsamples where functional and non-functional instances have matched prompt length distributions (binned matching, 10 quantile bins).

Table 9: AUC after controlling for prompt length. Average drop is 0.02.

	Original	Prompt-matched	Drop
JSON Schema			
LLaDA	0.80	0.75	0.052
Dream	0.82	0.80	0.022
GSM8K			
LLaDA	0.78	0.77	0.015
Dream	0.82	0.80	0.017
MBPP			
LLaDA	0.81	0.86	−0.046
Dream	0.88	0.85	0.028
ARC-Challenge			
LLaDA	0.73	0.70	0.030
Dream	0.71	0.68	0.029

J Entropy / Confidence Baseline

For each step we extract masked positions in the generation region and compute (a) mean entropy and (b) mean max-probability, pooled into the same 4 spatial regions as the probe. We evaluate three confidence signals: **Raw -entropy** ($-H$ pooled over regions), **Raw maxprob** (mean max-prob), and **LR-entropy** (logistic regression on the 8-dim feature vector, mirroring the probe's CV pipeline).

Table 10 reports the best per-step AUC across checkpoints for each method. The probe outperforms the strongest baseline in every (dataset,

model) setting, with an average gap of 0.097 AUC and a maximum gap of 0.179 on GSM8K-Dream. Table 11 reports filtering performance at $\tau = 0.80$ matching Table 2; the probe achieves higher classification accuracy than every confidence baseline on 7 of 8 settings (the exception is JSON-Dream, where LR-entropy reaches 78.3% but at less than half the compute saved).

Table 10: Best per-step AUC across checkpoints. Probe is the hidden-state logistic regression from §5.3; the other three columns use only the model's softmax distribution at each step.

	Probe	LR-ent	$-H$	maxprob
JSON Schema				
LLaDA	**0.846**	0.790	0.738	0.746
Dream	**0.827**	0.815	0.785	0.798
GSM8K				
LLaDA	**0.785**	0.701	0.675	0.667
Dream	**0.818**	0.639	0.646	0.646
MBPP				
LLaDA	**0.810**	0.730	0.724	0.725
Dream	**0.879**	0.776	0.749	0.744
ARC-Challenge				
LLaDA	**0.745**	0.611	0.623	0.610
Dream	**0.713**	0.588	0.592	0.580

Table 11: Filtering simulation at $\tau = 0.80$, formatted as compute-saved% / accuracy%. Probe column matches Table 2.

	Probe	LR-ent	$-H$	maxprob
JSON Schema				
LLaDA	98.4 / 73.5	41.0 / 61.0	80.3 / 52.2	82.1 / 54.4
Dream	96.7 / 72.4	43.1 / 78.3	85.8 / 50.0	83.1 / 52.2
GSM8K				
LLaDA	61.3 / 73.5	36.7 / 68.8	59.3 / 68.5	66.9 / 66.6
Dream	36.0 / 73.5	6.3 / 62.2	30.5 / 62.2	31.0 / 62.2
MBPP				
LLaDA	95.5 / 66.5	34.8 / 60.7	78.3 / 48.2	82.4 / 52.9
Dream	97.8 / 70.0	29.3 / 67.3	87.0 / 52.9	86.6 / 52.1
ARC-Challenge				
LLaDA	82.9 / 79.6	77.1 / 78.0	93.4 / 70.4	96.4 / 68.4
Dream	61.4 / 75.9	43.5 / 74.3	49.7 / 60.8	67.3 / 51.8

K Shallow-Feature Control

To test whether the hidden-state probe merely picks up surface cues, we extract per-task textual features from each generation and train a 5-fold cross-validated logistic regression on them ("shallow probe"). Features cover output length (character/line count), format markers (presence of expected delimiters such as ####, "‘`python`, code fences, JSON parseability, `def`), and, for ARC, one-hot indicators of the predicted answer letter (A/B/C/D). We compare the shallow probe's AUC to the best per-step hidden-state probe.

Table 12 summarizes the comparison. The hidden-state probe outperforms the shallow probe in 7 of 8 (dataset, model) settings, with an average gap of 0.12 AUC and a maximum gap of 0.25 on GSM8K-Dream. The single exception is ARC-Dream (-0.01), consistent with the marginal lift over majority baseline already noted for ARC in Table 2. On ARC, restricting the shallow probe to the predicted-letter one-hot alone yields AUC 0.640 for LLaDA and 0.681 for Dream, indicating that the answer letter is a partial but not dominant component of the shallow signal.

Table 12: Hidden-state probe vs. shallow-feature probe (logistic regression on output length, format markers, and ARC predicted letter). Probe column is the best per-step AUC across checkpoints; shallow is mean AUC over 5 folds.

	Probe	Shallow	Gap
JSON Schema			
LLaDA	**0.846**	0.693	+0.15
Dream	**0.827**	0.738	+0.09
GSM8K			
LLaDA	**0.785**	0.685	+0.10
Dream	**0.818**	0.567	+0.25
MBPP			
LLaDA	**0.810**	0.678	+0.13
Dream	**0.879**	0.673	+0.21
ARC-Challenge			
LLaDA	**0.745**	0.679	+0.07
Dream	0.713	**0.725**	−0.01